

Simulador didáctico de gestión de memoria con interfaz de autoaprendizaje basado en WWW

Félix Buendía, Julio Sahuquillo, Juan-Carlos Cano

Departamento de Informática de Sistemas y Computadores

Escuela Técnica Superior de Informática Aplicada

Universidad Politécnica de Valencia

e-mail: {fbuendia, jsahuqui, jucano}@disca.upv.es

Resumen

La enseñanza de las materias relacionadas con los Sistemas Operativos cubre un conjunto de contenidos esenciales en cualquier currículo universitario de informática. Las diferentes asignaturas impartidas, incluyen temas tales como la gestión de procesos y gestión de memoria, con una fuerte componente teórica. Aunque dichos conceptos son ampliamente tratados en numerosos libros de texto, no resulta sencillo encontrar entornos adecuados al nivel de conocimientos del alumno, que permitan realizar actividades donde poner en práctica los conceptos introducidos de forma teórica. En este trabajo, se describe una herramienta de simulación realizada de forma “*ad hoc*” para mostrar el funcionamiento de los principales aspectos relacionados con la gestión de memoria en un sistema operativo. Las características de dicha herramienta permiten que el alumno pueda interactuar con el simulador, introduciendo ejemplos de carga y configurando parámetros del sistema tales como el tamaño de la memoria y el algoritmo de gestión de la misma. La herramienta está desarrollada con tecnología de desarrollo basada en WWW lo que facilita su amplia difusión y utilización.

1. Introducción

Las asignaturas relacionadas con los Sistemas Operativos son un elemento fundamental en cualquier currículo universitario de informática. Desde las propuestas de ACM/IEEE de 1991 [1] hasta las más recientes del 2001 [2], existe un consenso sobre la importancia de esta temática. Las directrices del MEC establecen una serie de

descriptores que hacen referencia a aspectos como la gestión de procesos y ficheros así como los recursos de memoria y entrada/salida. Se trata de aspectos con una fuerte componente teórica pero que deben encontrarse apoyados con actividades prácticas. Sin embargo, no resulta sencillo encontrar entornos donde realizar este tipo de actividades, que permitan al alumno fijar con la práctica los conceptos tratados de forma teórica. Tradicionalmente, se han realizado actividades prácticas directamente sobre los sistemas operativos reales, donde el acceso a los mecanismos de gestión de recursos es complejo.

En algunos casos, se dispone de sistemas operativos reducidos como Nachos [4] que aunque presentan una amplia funcionalidad, para asignaturas troncales básicas tienen como principal inconveniente su orientación hacia aspectos de diseño e implementación.

Otras herramientas como RCOS [3] simulan el funcionamiento de un sistema operativo mediante animaciones. Sin embargo, la propuesta de RCOS se centra demasiado en los aspectos visuales y no profundiza en la aplicación de aspectos teóricos. En [5], Magee y Kramer presentan una herramienta que, de manera similar a la que se presenta en esta ponencia, se centra en la demostración visual de aspectos concernientes a aspectos relacionados con los conceptos de concurrencia.

En este trabajo, se presenta un simulador utilizado para mostrar el funcionamiento de la gestión de memoria de un sistema operativo. El simulador incorpora un entorno gráfico que visualiza la actividad del gestor de memoria. Este entorno junto con un menú detallado de ayuda facilita, sin duda, el autoaprendizaje de la materia al alumno.

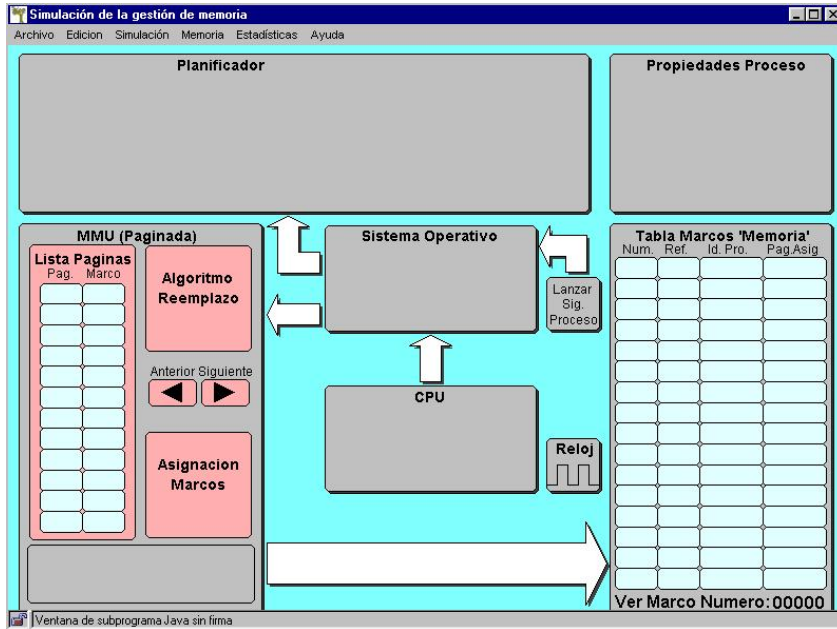


Figura 1. Esquema del simulador de gestión de memoria.

De hecho, son muchos los que durante el presente curso académico lo han utilizado para constatar los resultados de ejercicios propuestos en clase. Para facilitar esta tarea de aprendizaje, el simulador se encuentra disponible en [8].

El resto del trabajo se organiza como sigue. La Sección 2 describe la herramienta de simulación, mostrando su funcionamiento general. La Sección 3 detalla algunos ejemplos de ejercicios basados en la utilización de la herramienta. Finalmente, la Sección 4 presenta las conclusiones del trabajo.

2. Simulador de gestión de memoria

La herramienta de simulación ha sido desarrollada en lenguaje Java. Una primera versión de la misma dio lugar a un proyecto final de carrera [6], dirigido por el primer autor del

artículo y enmarcado dentro de la Escuela Técnica Superior de Informática Aplicada de la Universidad Politécnica de Valencia. Nuevas versiones con interfaz mejorada se han ido incorporando como parte del trabajo realizado en el ámbito de un proyecto europeo de educación a distancia [7]. La Figura 1 muestra la pantalla principal de la última versión de la herramienta, que es la que se utiliza para realizar prácticas en las asignaturas de Sistemas Operativos.

2.1. Introducción al simulador

El Simulador de gestión de memoria tiene por objeto mostrar el funcionamiento de algunos aspectos de la gestión de memoria que intervienen en la ejecución de uno o más procesos en un entorno de multiprogramación. Concretamente, la herramienta describe la ejecución de un conjunto

de procesos mediante el algoritmo de planificación *round-robin* y cómo estos procesos generan durante su ejecución accesos a direcciones lógicas que serán traducidas a direcciones en memoria física. El esquema de traducción de direcciones se basa en una técnica de paginación combinada con algoritmos de reemplazo, lo que se conoce globalmente como memoria virtual mediante paginación bajo demanda.

2.2. Inicio de una sesión

El primer paso en la ejecución de una simulación de gestión de memoria consiste en cargar el fichero de simulación (menú Archivo). Para describir un determinado programa, se ofrece un fichero denominado "Simulacion.inf" que describe las características de los diferentes procesos. La Figura 2 muestra un ejemplo.

```

Proceso1 50000 0
5000
0
100
0
456
4000
1050
end
0
Proceso2 50000 0
2048
7999
2248
3100
230
end
0
Proceso3 60000 0
2148
50000
3477
6000
2340
end EOF

```

Figura 2. Ejemplo de carga.

Este fichero contiene una secuencia de direcciones lógicas agrupadas por cada proceso que se incluye en el sistema. Para ello se utiliza una cabecera con tres campos: el primero hace referencia al identificador del proceso, el segundo el tamaño asignado a dicho proceso y el tercero a la prioridad, que identificará la cola a la que pertenece el proceso. Por ejemplo, la primera línea del fichero contiene la siguiente declaración:

```
Proceso1 50000 0
```

que define el Proceso1, con un tamaño de 50000 bytes y una prioridad 0. A partir de esta línea aparecen valores numéricos separados en diferentes líneas, cada uno de los cuales representa una dirección lógica. Las direcciones lógicas contenidas en un mismo proceso finalizan cuando aparece la etiqueta "end". A continuación, y antes de empezar la especificación de un nuevo proceso, aparece un valor numérico que especifica el desplazamiento temporal en la activación del siguiente proceso respecto al anterior. En este caso, se utiliza un valor 0 que indica que el Proceso1 y el siguiente (Proceso2) se activan al mismo tiempo. Un valor de 2 significaría que deben transcurrir dos instantes de tiempo antes que se active el Proceso2. La aparición de una etiqueta "EOF" a continuación de la última marca "end" indica la finalización del fichero de simulación.

Dentro de la opción "Ver Ventana de Direcciones Lógicas" (menú Edición) se puede editar el contenido del fichero de simulación, para modificar, por ejemplo, alguna de las direcciones lógicas o el tamaño de un proceso.

2.3. Configuración del sistema

Una vez que se ha cargado el fichero de simulación, se puede proceder a la configuración del sistema mediante una opción con dicho nombre del menú Archivo. La Figura 3 muestra un ejemplo. Se puede observar, como esta opción permite asignar diferentes atributos del sistema tales como el tamaño de la memoria, el tamaño de la página, los algoritmos de asignación y reemplazo del sistema.

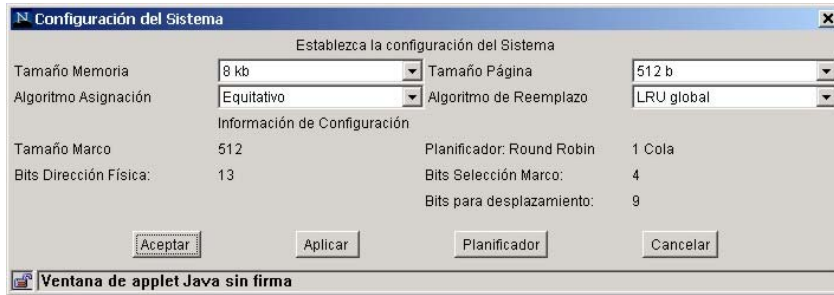


Figura 3. Ventana de configuración del sistema.

- **Tamaño de memoria:** indica el número de bytes disponibles en la memoria principal. El rango permitido oscila desde 8KBytes hasta 1Mbyte.
- **Tamaño de página:** indica la capacidad en bytes de cada una de las páginas en que se divide un proceso. Puede seleccionarse entre un rango de valores que van desde 512 bytes a 4 KBytes.
- **Algoritmo de reemplazo:** determina el método que se utiliza para seleccionar la página víctima de un proceso cuando se produce un fallo de página y no existe un marco de página disponible en memoria. Los posibles métodos a elegir son: LRU: local y global y FIFO: local y global.

En caso de utilizar un algoritmo de reemplazo local, habrá que elegir un determinado algoritmo de asignación, el cual determina el método utilizado para repartir los marcos de página disponibles en memoria entre los procesos del sistema. Existen tres métodos posibles:

- Equitativo.
- Proporcional.
- Prioritario.

La selección de atributos del sistema permite determinar aspectos tales como el tamaño del marco de página, el número de bits de la dirección física y cómo estos se distribuyen (los bits de mayor peso se destinan a la selección del marco y los de menor peso al desplazamiento dentro del marco).

Por otra parte, tal y como se muestra en la Figura 3, mediante la opción “Planificador” también pueden configurarse ciertos parámetros referidos a la Planificación del Sistema. Los parámetros que se pueden configurar son los siguientes:

- **Algoritmo de planificación:** actualmente, sólo se encuentra implementado un algoritmo de planificación del tipo “round-robin” (turno rotatorio). Cabe matizar, que existen 2 proyectos fin de carrera en fase de escritura destinados a implementar otros algoritmos.
- **Número de colas:** indica el número de estructuras de datos para almacenar procesos activos. Se pueden elegir de 1 a 5 colas mediante los identificadores cola 0 a cola 4, donde la cola 0 es la más prioritaria.
- **Quantum:** permite definir para cada cola el “quantum” o cantidad de tiempo que se asocia a un proceso en su turno de ejecución. En este caso, dicho tiempo se evalúa en base al número de direcciones lógicas. Por defecto su valor es 1, es decir, se accede a una dirección lógica en cada instante de tiempo.

2.4. Comienzo de la simulación

Una vez cargado el fichero de simulación y realizada la configuración del sistema, para simular el comportamiento se elige la opción “Comenzar Simulación” dentro del menú de Archivo.

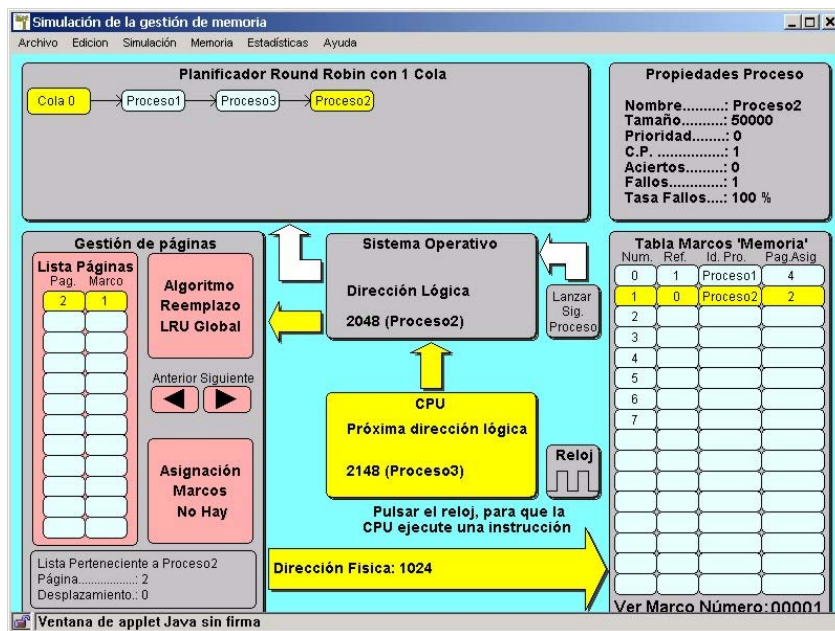


Figura 4. Ejemplo de ejecución del simulador.

Una vez iniciada la simulación, se puede observar la evolución de la misma a través de diferentes zonas de la pantalla mostradas en la Figura 4. Existen dos formas de controlar la simulación, las cuales se pueden seleccionar desde el menú de Simulación:

- Modo paso a paso: permite visualizar los resultados de cada paso (unidad temporal) en que se realiza cada acceso a memoria o la activación de un nuevo proceso. Para ello, el usuario debe pulsar el icono de Reloj que aparece en la parte central de la pantalla.
- Modo automático: en este caso la ejecución de los procesos y por tanto, el acceso a direcciones de memoria se realiza de forma automática sin intervención del usuario.

La parte superior izquierda de la pantalla muestra una representación gráfica de los

procesos activos en cada cola. En el ejemplo se utiliza una única cola (Cola 0) que contiene los tres procesos definidos. Además, cada vez que se accede a la dirección lógica de un proceso, éste se pondrá al inicio de la cola y cambiará el color del símbolo que lo representa, para indicar que es el proceso en ejecución.

La parte superior derecha de la pantalla informa del estado del proceso en ejecución. Para ello, se incluye el contador de direcciones lógicas (C.P.) que define el orden relativo de la dirección lógica accedida por el proceso en ese momento y que se actualiza con cada nuevo acceso. Además, se indica el número de accesos realizados hasta el momento actual, diferenciando entre fallos de página (la página que contiene la dirección actual no está en memoria principal y tiene que ser transferida desde disco) y aciertos (la página accedida está ya en memoria). También se incluye el valor de la tasa de fallos.

La parte inferior izquierda representa la gestión de páginas e incluye información como:

- El algoritmo de reemplazo seleccionado.
- El tipo de asignación de marcos si lo hubiera.
- La lista de páginas utilizada por el correspondiente algoritmo de reemplazo para elegir la página víctima y decidir el marco que va a ser liberado. Esta estructura incluye unos iconos (“Anterior” y “Posterior”) para poder manejar los elementos de la lista que excedan el tamaño dispuesto en la pantalla (12 páginas). También se incluye información de la dirección lógica actual, como la página y el desplazamiento dentro de ella. Esta lista no representa la tabla de páginas ya que sólo almacena parte de su información (en este caso el número de marco asociado a una página).

La parte inferior derecha de la pantalla representa la Tabla de Marcos de la memoria física. En ella se muestra la siguiente información:

- El número de marco: su valor va desde 0 hasta N, siendo N+1 el número de marcos disponible. En caso de sobrepasar el número máximo de elementos de la tabla en pantalla (16), se podrán utilizar una serie de controles que se activan al colocar el cursor en alguno de los números de la opción “Ver Marco Número”. Cuando se activan estos controles aparece el símbolo de una mano y al pulsar el botón izquierdo del ratón el número se incrementa, mientras que al pulsar el botón derecho, se decrementa. Ello permite el acceso a la totalidad de la estructura de la tabla de marcos de página.
- El campo referencia define la antigüedad de las páginas que se almacenan en memoria y sirve para decidir el marco ocupado por la página a reemplazar, en caso de fallo. Por tanto, depende del tipo de algoritmo de reemplazo elegido.
- El identificador de proceso: indica el proceso que accede a la página.
- Página asignada: se trata de la página almacenada en el marco referido.

Por último, la parte central de la pantalla define la relación entre las estructuras descritas previamente. Indica en cada momento, que

proceso ha sido elegido por parte del planificador para su ejecución y la dirección lógica implicada en este paso. Dicha dirección puede ser seleccionada de forma automática (modo “Automático”) o interactiva a través del icono Reloj (modo “Paso a paso”). Asimismo, se puede activar un nuevo proceso mediante el control “Lanzar Siguiente Proceso”, sin que se tenga que esperar al plazo de tiempo definido en el fichero de carga.

También se encuentran en esta parte de la pantalla tres etiquetas que contienen la siguiente información:

- Sistema operativo: indica la dirección lógica que actualmente está siendo traducida.
- CPU: representa la próxima dirección a traducir.
- Dirección física: contiene la dirección física resultado de la traducción actual.

2.5. Resultados de la simulación

Mediante la opción “Ver Ventana de Direcciones Físicas” del menú Edición, se obtiene una traza de la simulación. Esta traza muestra un listado de direcciones físicas como consecuencia del proceso de traducción. La Figura 5 muestra un ejemplo de dicha traza.

```
*****
Proceso1 Dir.Lógica: 5000 Dir. Física: 904
Proceso2 Dir. Lógica: 2048 Dir. Física: 1024
Proceso3 Dir. Lógica: 2148 Dir. Física: 2148
Proceso1 Dir. Lógica: 0 Dir. Física: 3072
Proceso2 Dir. Lógica: 7999 Dir. Física: 4927
Proceso3 Dir. Lógica: 50000 Dir. Física: 5968
Proceso1 Dir. Lógica: 100 Dir. Física: 3172
Proceso2 Dir. Lógica: 2248 Dir. Física: 1224
Proceso3 Dir. Lógica: 1024 Dir. Física: 6144
Proceso1 Dir. Lógica: 0 Dir. Física: 3072
Proceso2 Dir. Lógica: 4096 Dir. Física: 7168
Proceso3 Dir. Lógica: 1024 Dir. Física: 6144
Proceso1 Dir. Lógica: 456 Dir. Física: 3528
Proceso2 Dir. Lógica: 2134 Dir. Física: 1110
Proceso3 Dir. Lógica: 3477 Dir. Física: 405
Proceso1 Dir. Lógica: 4000 Dir. Física: 2976
Proceso2 Dir. Lógica: 3100 Dir. Física: 4124
Proceso3 Dir. Lógica: 6000 Dir. Física: 6000
Proceso1 Dir. Lógica: 1050 Dir. Física: 7194
*****
```

Figura 5. Ventana de direcciones físicas

Tras finalizar la simulación y descargar los procesos de la memoria, se pueden obtener estadísticas de la ejecución de la simulación. La Figura 6 muestra un ejemplo de dichos resultados, los cuáles se pueden obtener utilizando las opciones “Tasa de Fallos” y “Rendimiento” dentro del menú “Estadísticas”.

```

*****
Tasa de Fallos
*****
El Proceso Proceso3 ha efectuado 7 accesos a memoria
obteniendo 1 aciertos y 6 fallos de página
Con una tasa de fallos del 85 %

El Proceso Proceso2 ha efectuado 7 accesos a memoria
obteniendo 2 aciertos y 5 fallos de página
Con una tasa de fallos del 71 %

El Proceso Proceso1 ha efectuado 7 accesos a memoria
obteniendo 3 aciertos y 4 fallos de página
Con una tasa de fallos del 57 %
*****
Rendimiento del sistema
*****
Se han producido un total de 21 accesos a memoria
contabilizando un total de 6 aciertos de página y
15 fallos de página. Con una tasa de fallos del 71 %

```

Figura 6. Estadísticas de la simulación.

3. Aplicación del simulador a las prácticas de Sistemas Operativos

En la sesión práctica relacionada con la gestión de memoria, se sugiere al alumno que siga los pasos mencionados en el apartado anterior. A partir de un fichero de simulación, el alumno debe configurar el sistema de varias formas interpretando los resultados obtenidos. El alumno debe realizar las siguientes tareas.

- Cargar el fichero de simulación.
- Configurar el sistema. Se sugiere el uso de un tamaño de memoria de 8 KB, un tamaño de página de 1 KB y alguno de los algoritmos de reemplazo global (LRU o FIFO). Respecto al planificador, se puede considerar al principio, una única cola (cola 0).
- Comenzar la simulación. Para un mejor control, se puede utilizar un modo “paso a

paso” y la visualización simultánea de los resultados en la “Ventana de Direcciones Físicas”. Se sugiere al usuario que antes de acceder a la siguiente dirección lógica, analice la información disponible y determine el proceso y si provocará un fallo de página, comprobando posteriormente qué página ha sido reemplazada y la dirección física generada.

- Evaluar los resultados de la simulación. Una vez finalizada la simulación se trata de comprobar los resultados obtenidos que reflejan la tasa de fallos de cada proceso y el rendimiento global del sistema.

Con la información obtenida se plantean ejercicios relacionados con la práctica realizada, y cuyo objetivo es analizar el nivel de adquisición de conocimientos del alumno. A continuación se muestra un posible ejercicio:

- Determinar el estado final de la memoria según una configuración de tamaño de página de 1 KB, algoritmo LRU de reemplazo GLOBAL y tamaño de memoria de 8 KB. Para especificar la solución, se le ofrece al alumno la tabla adjunta.

Marco	Pid	Pag.
0		
1		
2		
3		
4		
5		
6		
7		

Tabla 1. Ejemplo de ejercicios basados en tablas.

Aunque dicho ejercicio, es relativamente “mecánico” de utilización del simulador, también se sugiere al alumno que previamente se realice de forma completamente manual, de forma que el alumno pueda contrastar su conocimiento de la materia, utilizando el simulador para solucionar posibles dudas. También pueden plantearse actividades de tipo reflexivo que permitan validar el nivel de conocimientos y el trabajo del alumno.

A continuación se muestran algunos ejemplos de estos ejercicios propuestos al alumno.

- Indique si existe alguna diferencia en la aplicación de ambos algoritmos (LRU y FIFO) y en qué consiste. Justifique la existencia de una misma tasa de fallos de páginas en ambos ejemplos.
- Encuentre un ejemplo de carga donde la aplicación de un algoritmo LRU devuelva un menor número de fallos de página, respecto un algoritmo FIFO.

4. Conclusiones

En este trabajo, se ha presentado una herramienta dirigida a la simulación de algoritmos de gestión de memoria. Dicha herramienta ha sido realizada en lenguaje Java, lo que facilita su utilización en entornos Web y su portabilidad a diferentes sistemas operativos. Asimismo, proporciona una interfaz visual que favorece su uso por parte de los alumnos. Tras tres años de experiencia los alumnos han aceptado de buen grado y manifestado la utilidad de dicho simulador como herramienta de autoaprendizaje.

La herramienta, además de reproducir información en un formato atractivo, invita al alumno a interactuar con la aplicación y permite fácilmente la modificación de algunos de los parámetros de entrada como el tamaño de la página o de la memoria principal. La herramienta es de gran utilidad para el alumno principalmente para la comprobación de ejercicios planteados y resueltos en clase. La experiencia, en líneas generales, ha resultado positiva a pesar del elevado coste de desarrollar una herramienta *ad hoc* de uso específico.

Como trabajo en vías de realización, se está considerando la integración de:

- Algoritmos de planificación basados en prioridades.
- Aspectos teóricos en el funcionamiento del simulador
- Algoritmos de reemplazo de páginas alternativos.

Referencias

- [1] A.B. Tucker et al., ACM/IEEE-CS Joint Curriculum Task Force. Computing Curricula 1991, ACM Press; IEEE Comp. Soc. Press. , 1991.
- [2] ACM/IEEE Task Force on the Year 2001 Model Curricula for Computing , Computing Curricula 2001(CC-2001), obtenido el 10/11/1999 de la página web: <http://www.computer.org/education/cc2001/>.
- [3] David Jones, Andrew Newman, A constructivist-based tools for operating systems education, Proceedings of EdMedia'2002, Denver, Colorado, June 2002.
- [4] W A Christopher et a. (1993), The Nachos Instructional Operating System. Proceedings of the Winter 1993 Usenix Technical Conference, pp 481-489.
- [5] Jeff Magee and Jeff Kramer. Concurrency: State Models & Java Programs.
- [6] Llopis Mengual, J.Espinosa Rodilla. Simulador de gestión de memoria. Proyecto final de carrera (PFC1), 1999.
- [7] Innovations for Education in Information Technology through Multimedia and Communication Networks. Proyecto Socrates de Red Temática. <http://www.eui.upv.es/ineit-mucon/>.
- [8] Programa de Simulación de Gestión de memoria. http://www.eui.upv.es/ineit-mucon/Applets/mem_simulator/Applet.htm.